IMPROVING THE CHARGE SIMULATION METHOD FOR THE COMPUTATION OF HIGH VOLTAGE ELECTRIC FIELDS WITH EFFICIENT LEAST SQUARES TECHNIQUES

J. N. Hoffmann
Companhia Paranaense de Energia (COPPEL) - P.O.Box 318
80.020.000 - Curitiba, PR, Brazil
fax nr. 55-41-331-3115

P. Pulino
Universidade Estadual de Campinas (UNICAMP) - P.O.Box 6065
13.081.000 - Campinas, SP, Brazil
fax nr. 55-19-239-5808

Abstract: The well known Charge Simulation Method, which is commonly used for electric field calculations, is shown to be a particular and ill-conditioned case of the Least Squares Charge Simulation Method. By solving a practical problem, it is shown how to efficiently handle a least squares problem, thus obtaining results of higher precision if compared to the traditional Charge Simulation Method. For the solution of the resulting linear system, several mathematical methods are analyzed and compared, being stated that the optimum combination of higher precision, lesser error propagation, lesser CPU time and lesser computer RAM are simultaneously reached when applying the Least Squares Charge Simulation Method, solved with the QR decomposition and Householder transformations.

Key words: Electric fields, numerical methods, high voltage engineering, electrical engineering computing, digital simulation, least squares methods.

1. INTRODUCTION

The Charge Simulation Method (CSM) has been very commonly used for electric field computations in the last 20 years. From its introduction by [1], the method was modified with optimization techniques (OCSM) [2], least squares techniques (LSCSM or LSEM) [3], and in combination with other numerical methods (finite elements and finite differences) [4,5]. References [6,7] also present a good overview and some applications of the method.

While the mathematical formulation of the OCSM is somewhat more complicated, the CSM and the LSCSM reduce to few and relatively easy steps:

- Choose arbitrary points over the boundaries where electric potentials are known (vector \( \Phi \)),
- Choose arbitrary electric charges placed inside the given electrode (vector \( q \)),
- Solve the system of linear equations \( \mathbf{P} \mathbf{q} = \mathbf{\Phi} \) then obtained,
- Calculate the electric potential \( \mathbf{\Phi} \) and the electric field vector \( \mathbf{E} \) where desired, using the solution vector \( \mathbf{q} \).

\( \mathbf{P} \) is a full matrix obtained from known relations between points and charges (the Maxwell Electric Potential Coefficients [1]), which depend essentially on geometric data. In the CSM, the number of contour points is equal to the number of simulated charges, while in the LSCSM a lesser number of simulated charges is chosen.

As shown above, the CSM and the LSCSM are simple to apply. In some cases, however, the CSM lead to poor results (see section 2). On the other hand, if the LSCSM is used with the so called normal equations (see [3], Appendix 3), it may result in a system of linear equations with a condition number (see Appendix V) many times higher that the CSM itself. In this case the increased error propagation may also lead to poor results.

Considering these points, the present work is intended to show how to deal with these methods in a more efficient way. The result is that the LSCSM, if properly applied, presents incontestable advantages over the traditional CSM.

Section 3 is devoted to a detailed error analysis, stating the optimum relation number of charges / number of contour points. Section 4 deals with the problem from the condition number point of view. Section 5 presents the QR Decomposition and the Singular Value Decomposition methods for the solution of the LSCSM, allowing the resulting linear system a condition number many times smaller than the CSM or than the LSCSM with normal equations. In addition, nine different mathematical methods are analyzed and compared for the solution of the problem. Appendices I and V include the mathematical background necessary for a full understanding of the problem.

Based on the results obtained with the new proposed formulation for the LSCSM, it can be concluded that, with an adequate relation number of charges / number of contour points, and with an adequate mathematical and computational treatment, the LSCSM results remarkably more efficient than the CSM. As a matter of fact, the LSCSM constitutes the generalization for the methods of simulated charges, while the traditional CSM is a particular and ill-conditioned case.

2. APPLICATION

Consider the three-dimensional axi-symmetric problem of Figure 1, to be solved with the CSM and the LSCSM. The same problem was also solved by [6], using the CSM only. Note that problem 2 is similar to problem 1, except that it includes different electrical permittivities. The simulated (ring) charges and contour points were considered as shown in Table I.

<table>
<thead>
<tr>
<th>Table 1 - Number of Charges and Contour Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Problem 1</td>
</tr>
<tr>
<td>-----------</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Electrode</td>
</tr>
<tr>
<td>Enclosure</td>
</tr>
<tr>
<td>In dielectric near air</td>
</tr>
<tr>
<td>In air, near dielectric</td>
</tr>
<tr>
<td>Total</td>
</tr>
<tr>
<td>Order of the linear system</td>
</tr>
<tr>
<td>Order of the linear system</td>
</tr>
</tbody>
</table>

A detailed formulation of the CSM equations for problems 1 and 2 is found in [6]. For both cases, a systems of m linear equations and n = m unknown electric charges is obtained, which can be written as
Figure 1 - Three-dimensional axi-symmetric application

\[ F \cdot q = \Phi, \]  

(1)

where

- \( F \) is a full square matrix of order \( n \times n \),
- \( q \) is a vector of order \( n \) with the values of simulated charges which represent the physical problem,
- \( \Phi \) is a vector with the given electric potentials.

The same procedure is applicable to the LSCSM, except that a smaller number of simulated charges is chosen, resulting in a system of linear equations with \( m \) equations and \( n \) unknown electric charges (\( m > n \)). This system can also be written in the condensed form (1), where \( F \) is now a rectangular matrix, of order \( m \times n \).

A comparison of various methods for the computation of \( q \) is found in section 5. Once \( q \) is obtained, the electric potential \( \Phi \) and the electric field \( E \) can be computed at every desired point \( P \), by means of

\[ \Phi(P) = \sum_{j=1}^{n} p_j q_j \]  

(2)

\[ E(P) = \sum_{j=1}^{n} \left( f_{ij} \Delta r + f_{ij} \Delta z \right) q_j \]  

(3)

where

- \( p_j \) and \( f_{ij} \) are Maxwell coefficients (see [1,6]),
- \( q_j \) are the simulated charges obtained from \( q \), and \( \Delta r \) and \( \Delta z \) are unit vectors for directions \( r \) and \( z \).

An exact solution would give the equipotential line of \( \Phi = 1.0 \) for all \( P \) located on the electrode boundary. However, due to the finite discretization of charges and contour points, the CSM and the LSCSM give an equipotential line which deviates somewhat from the electrode boundary, as shown by Figure 2.

It is clearly shown in this figure the surprising fact that a reduction in the number of simulated charges gives better results, since the respective equipotential line of \( \Phi = 1.0 \) clearly fits better to the electrode contour. The error of each method is quantified in section 3.

In Figure 3, the electric potentials and electric field as computed by [6] for gap A-B of Figure 1 are reproduced and compared to the values computed with the LSCSM with data taken from Table 1. Note that the previous unstable zone

![Figure 3 - Field calculation with CSM and LSCSM for gap A-B](image-url)
generated by the CSM (strong oscillations) is not present in the computation with the LSCSM. Furthermore, in [6] matrix $P$ required the calculation of $77 \times 77 = 5929$ Maxwell coefficients, while the LSCSM used here required $38 \times 38 = 2728$, what means considerable reduction in CPU time, as well as in required computer RAM.

It can be concluded that the CSM is more unstable and gives poorer results if compared to the LSCSM, particularly close to the boundaries. This is due to the fact that matrix $P$, having logarithm terms or similar, is close to a singular (non invertible) matrix and therefore, ill-conditioned (see Appendix V).

3. ERROR ANALYSIS

The conclusions of section 2 can be quantified by plotting the cumulative error of the electric potentials computed along the electrode boundary for several relations $n/m$, as shown in Figure 4. For instance, if we take the LSCSM with $n/m = 0.90$, the electric potentials computed along the electrode boundary (where the exact solution is $\phi = 1.0$) have an error greater than 1 % for 50 % of the electrode contour length.

4. CONDITION NUMBER ANALYSIS

The condition number of the linear system (1) depends on the relation $n/m$. Figure 6 shows the behavior of this parameter, which can be computed with the SVD method (see Appendix III.3).

It is clear from Figure 6 that the method used for the solution of the LSCSM equations is of primal importance. The LSCSM equations solved by the method proposed in [31] (normal equations) may generate a matrix $P$ with a condition number greater than the CSM itself, what may lead to severe error propagations during the computations.

On the other hand, the LSCSM equations solved by the QR decomposition (Appendix III.2) or the SVD (Appendix III.3) always result in a condition number for $P$ which is lower than the condition number computed for the CSM and for the LSCSM with normal equations. Therefore, as far as the condition number is concerned, the QR decomposition and the SVD methods are more effective.
5. METHODS FOR THE SOLUTION OF
THE RESULTING LINEAR SYSTEM

It can be concluded from above sections that the LSCSM,
with an adequate choice for the relation n/m, gives results
higher precision than the straight CSM. It was also shown
that under the condition number point of view, the QR
decomposition and the SVD methods are preferred to the
method of normal equations used so far.

In this section the problem is analyzed as function of
the CPU time and the required computer RAM, including a good
number of mathematical methods for the computation of the
vector of charges $q$.

Note that the CPU computer time is also a good indication
of the number of arithmetic operations taken by the
computer. Thus, the bigger is the required CPU computer time
the bigger is the possibility of a significant error
propagation.

Table 2 shows the CPU time required by an
IBM 3090-300S computer for the solution of problems 1 and 2,
taken in percentile rates of the time required for the CSM.

<table>
<thead>
<tr>
<th>TABLE 2 - COMPARISON OF VARIOUS METHODS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method of</td>
</tr>
<tr>
<td>calculation</td>
</tr>
<tr>
<td>-------------</td>
</tr>
<tr>
<td>CSM</td>
</tr>
<tr>
<td>LU</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>LSCSM</td>
</tr>
<tr>
<td>(with n/m=1)</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>SVD</td>
</tr>
</tbody>
</table>

From Table 2 it can be concluded that

a) The LSCSM solved with the QR decomposition and
Householder transformations presented the best performance,
since it requires lesser CPU computer time and lesser RAM.

b) The CGS method, which showed an excellent performance
in problems with sparse matrices as reported by [5], did not
show a good performance here with a full matrix.

c) All methods for the solution of the LSCSM with normal
equations (i.e. CG, CGS, CD and LU) require special
computational considerations in order to avoid additional
RAM required for the storage of matrix $P^TP$.

6. CONCLUSIONS

The traditional Charge Simulation Method was shown to be
a particular and ill-conditioned case of the Least Squares
Charge Simulation Method, which is the generalization.

The best relation number of charges / number of contour
points (n/m) lies in the range 0.35 < n/m < 0.80, where the
LSCSM results remarkably more precise than the straight CSM,
specially close to the boundaries of the problem.

The LSCSM computed with the QR decomposition generates a
system of linear equations with a condition number
significantly lower than the LSCSM with normal equations as
used so far, thus reducing error propagation during the computations.

The optimum combination of higher precision, lesser error
propagation, lesser CPU computer time and lesser computer
RAM are simultaneously reached when applying the Least
Squares Charge Simulation Method, solved with the QR
decomposition and Householder transformations.

APPENDIX I

Mathematical background (I(8,9,10,11,12))

Let $A = [a_{ij}]$ denote a real matrix of order $m x n$ with elements $a_{ij}$, and $x = [x_1, x_2, ..., x_n]^T$ denote a real column vector of order $n$ with elements $x_i$.

The transpose of $A$ is $A^T$, where $A^T = [a_{ji}]$. $A$ is symmetric if $A^T = A$. If $m=n$, the inverse of $A$ is denoted by $A^{-1}$, with $A^{-1}A = I$ (unit matrix).

The Euclidean norm of vector $x$ is $\|x\| = (\sum x_i^2)^{1/2}$. Another usual vector norm is the infinite norm $\|x\|_\infty = \max |x_i|$. A norm for matrix $A$ can be defined as $\|A\| = \max \|Ax\|/\|x\|$, for all $x \neq 0$.

The n column vectors $a_i$ from $A$, of order $m$, is said to be linearly independent if $\sum a_i x_i = 0$ is satisfied only for $x_i = 0$. In this case, $A$ is said to be of rank $n$. $A$ is invertible if $\text{rank}(A) = m = n$.

If $x^T A x > 0$ for all vector $x \neq 0$, $A$ is said to be positive definite. If $\text{rank}(A) = n$, $A^T A$ is symmetric and positive definite.

If $A^T A = I$, $A$ is said to be orthogonal and in this case, $\|A\| = 1$. $A$ is a lower triangular matrix if $a_{ij} = 0$ for all $i > j$, upper triangular if $a_{ij} = 0$ for all $i < j$, and diagonal if $a_{ij} = 0$ for all $i \neq j$.

The inner product of two vectors $x$ and $y$ is a real number denoted by $(x,y) = \Sigma x_i y_i$.

APPENDIX II

Matrix decompositions

LU decomposition: If $\text{rank}(A) = m = n$, then only one decomposition $A = LU$ exists, where $U$ is upper triangular and $L$ is lower triangular, with unit diagonal (I(8,9,10)).

QR decomposition: If $\text{rank}(A) = n$, then only one decomposition $A = QR$ exists, where $R$ is upper triangular of order $n x n$ and $Q$ is orthogonal of order $m x n$ (I(8,9,12)).

Singular Value decomposition (SVD): $A$ can be written as $A = U S V^T$, where $U$ is orthogonal of order $m$, $V$ is orthogonal of order $n x n$ and $S$ is diagonal of order $n x n$, whose diagonal elements $s_1 \geq s_2 \geq \cdots \geq s_n \geq 0$ are known as the singular values of $A$ (I(8,9,11,12)).

Cholesky decomposition: If $A$ is symmetric and positive definite, then only one decomposition $A = R^T R$ exists, where $R$ is upper triangular with positive elements in the main diagonal (I(8,10,12)).
APPENDIX III

Overdetermined linear systems

Let \( A \mathbf{x} = \mathbf{b} \) be a linear system of \( m \) equations and \( n \) unknown values, so that \( m > n \). This system of equations is said to be overdetermined, and usually have no exact solution. Let \( \mathbf{r} = \mathbf{b} - A \mathbf{x} \) be the residual vector.

It is reasonable to search for an approximate solution to this linear system, giving the “lowest possible residual vector.” Usually a least squares method is considered, which consists in determining the unique vector \( \mathbf{x} \) which gives the lowest Euclidean norm for the residual vector (minimum \( \| \mathbf{r} \|^2 \)). A brief description of the most common methods follows.

III.1 Method of the normal equations

It is shown by [8,9,12] that vector \( \mathbf{x} \), solution for the least squares problem, is also the solution of the linear system

\[
A^T A \mathbf{x} = A^T \mathbf{b} \tag{4}
\]

of order \( n \times n \), also known as a system of normal equations.

Since \( A^T A \) is symmetric and positive definite, some mathematical procedures are applicable (see Appendix IV.1) for the solution of (4). On the other hand, the condition number of the resulting linear system is affected since \( |k(A^T A)| = |k(A)|^2 \), which may compromise the solution due to increase of error propagation.

III.2 Method of the QR decomposition

It is shown by [8,9,12] that vector \( \mathbf{x} \), solution for the least squares problem, is also the solution of

\[
R \mathbf{x} = Q^T \mathbf{b} \tag{5}
\]

where \( QR = A \). This linear system of order \( n \times n \) is easily solved by back-substitution since \( R \) is upper triangular. Therefore, the main computational effort is the determination of the QR decomposition of \( A \) (Appendix IV.2).

III.3 Method of the Singular Value decomposition

It is shown by [8,9,11,12] that vector \( \mathbf{x} \), solution for the least squares problem, is given by

\[
\mathbf{x} = V \Sigma^{-1} U^T \mathbf{b} \tag{6}
\]

where \( U \Sigma V^T = A \) is the Singular Value decomposition of \( A \). Since \( S \) is a diagonal matrix, \( S^{-1} \) is easily obtained and then expression (6) reduces to simple matrix-vector products. As in section III.2, the main computational effort is the determination of the Singular Value decomposition of \( A \) (Appendix IV.3).

APPENDIX IV

Mathematical methods for least squares problems

IV.1 Methods for the solution of the normal equations (equation 4)

The conjugate gradients (CG), the conjugate gradients squared (CGS), the Cholesky decomposition (CD) and Gauss pivoting (LU decomposition) methods are analyzed in the following sections.

IV.1.1 Conjugate gradients (CG)

The CG method is only applicable to symmetric and positive definite linear systems, as (4). The algorithm is demonstrated by [8]. The matrix product \( A^T A \) is avoided using the property of inner products \( \langle A^T \mathbf{A} \mathbf{x}, \mathbf{y} \rangle = \langle \mathbf{A} \mathbf{x}, \mathbf{A} \mathbf{y} \rangle \), which is easily demonstrated. The resulting algorithm for the solution of (4) is

\[
x_0 = 0
\hat{r}_0 = A^T b
\text{For } k = 1, \ldots n:
\]

\[
\begin{align*}
& \text{if } \hat{r}_{k-1} = 0 \quad \text{then} \\
& \quad \text{Set } x = \hat{x}_{k-1} \text{ and quit} \\
& \text{else} \\
& \quad \beta_k = \langle \hat{r}_{k-1}, \hat{r}_{k-1} \rangle / \langle \hat{r}_{k-1}, A \hat{r}_{k-1} \rangle \\
& \quad \alpha_k = \langle \hat{r}_{k-1}, \mathbf{b} \rangle / \langle \mathbf{b} \rangle \\
& \quad \hat{x}_k = \hat{x}_{k-1} + \alpha_k \hat{r}_{k-1} \\
& \quad \hat{r}_k = A^T (b - A \hat{x}_k)
\end{align*}
\]

IV.1.2 Conjugate gradients squared (CGS)

The CGS doesn’t require \( A \) to be symmetric and positive definite, unlike the CG method. It is included here due to the good results obtained with sparse matrix as reported by [5,13]. The final algorithm [13] for the solution of (4) is

\[
\begin{align*}
& \text{estimate } x_0 \\
& \hat{r}_0 = A^T (b - A x_0) \\
& q_0 = p_{-1} = 0; \quad \rho_{-1} = 1; \quad n = 0 \\
& \text{while } n < \text{tolerance} \text{ do} \\
& \quad \rho_n = \langle \hat{r}_n, p_n \rangle; \quad \beta_n = \rho_n / \rho_{n-1} \\
& \quad u_n = \hat{r}_n + \beta_n q_n \\
& \quad p_n = u_n + \beta_n (q_n - \beta_n p_{n-1}) \\
& \quad v_n = A^T (Ap_n) \\
& \quad s_n = \langle \hat{r}_n, v_n \rangle; \quad \alpha_n = \rho_n / s_n \\
& \quad q_{n+1} = u_n - \alpha_n v_n \\
& \quad x_{n+1} = x_n + \alpha_n (u_n - q_{n+1}) \\
& \quad \hat{r}_{n+1} = A^T (b - A x_{n+1}) \\
& \quad n = n+1
\end{align*}
\]

IV.1.3 Cholesky decomposition (CD)

Since \( A^T A \) of (4) is symmetric and positive definite, it is possible to use the Cholesky decomposition \( A^T A = R^T R \) (where \( R \) is upper triangular). Therefore, equation (4) becomes \( R^T R \mathbf{x} = A^T \mathbf{b} \). The later is easily solved with an auxiliary vector \( y \), according to the sequence

\[
R^T y = A^T \mathbf{b} \quad (y \text{ is obtained by forward substitution})
\]

\[
R \mathbf{x} = y \quad (x \text{ is obtained by back substitution})
\]

Note that the matrix product \( A^T A \) needs to be computed before \( R \). The elements \( r_{ij} \) of \( R \) are computed from the elements of matrix \( A^T A = A \cdot A \cdot y \), with the algorithm (112)

\[
\text{For } j = 1, \ldots n \\
\text{For } k = 1, \ldots j-1 \\
\quad r_{kj} = R_{kj} - \sum_{i=k}^{j-1} r_{ki} r_{ij} / r_{kk} \\
\text{end}
\]

\[
\text{end}
\]

IV.1.4 Gauss pivoting (LU decomposition)

The well known method of Gauss pivoting [8,9,10,11,12] can be used for the computation of the decomposition \( A^T A = LU \) (Appendix II). Thus, equation (4) becomes \( L U \mathbf{x} = A^T \mathbf{b} \), which is easily solved with an auxiliary vector \( y \), according to the sequence

\[
L y = A^T \mathbf{b} \quad (y \text{ is obtained by forward substitution})
\]

\[
U \mathbf{x} = y \quad (x \text{ is obtained by back substitution})
\]

As in section IV.1.3, the matrix product \( A^T A \) is computed before the LU decomposition .

101
IV.2 Methods for the QR decomposition (for equation 5)

The Classical Gram-Schmidt, Modified Gram-Schmidt, Givens rotations and Householder transformations methods are briefly described in the following.

IV.2.1 Classical Gram-Schmidt (CLG)

In this method, matrices Q and R are sequentially computed column by column. The algorithm and detailed mathematical description are found in [8,9,12]. Once A is overwritten by Q, an additional space of dimension \( \frac{n(n+1)}{2} \) is necessary for the storage of R.

IV.2.2 Modified Gram-Schmidt (MGO)

This is a slight modification to the CLG method, leading to a better numerical performance. Q is computed column by column, and R is computed line by line. Other comments as for CLG. See [8,9,12] for algorithm and details.

IV.2.3 Givens rotations (GR)

Algorithm and mathematical description are found in [8]. A is overwritten by R, and the product \( Q^T \) (necessary for equation 5) is computed while computing R, thus not requiring additional RAM for R as the CLG or MOG methods.

IV.2.4 Householder transformations (HT)

A matrix H of order \( n \times n \), defined as

\[ H = I - 2 \frac{vv^T}{v^Tv} \]

is said to be a Householder transformation (also known as Householder matrix or Householder reflection), where v is a vector of dimension n. It can be shown that H is an orthogonal matrix [8].

In this method, an adequate H is chosen so that the product \( HA \) results in a matrix with null elements below the main diagonal of A at a given column. By a chain repetition of this product we get the matrix \( R = HA...H_2H_1A = Q^T \)

Algorithm and mathematical description are found in [8,12].

As in the GR method, A is overwritten by R, and the product \( Q^T \) (necessary for equation 5) is computed sequentially while computing R. Therefore, no additional RAM is necessary for R.

IV.2 Methods for the Singular Value decomposition (for equation 6)

As shown in section III.3, the solution for a least squares problem by the SVD reduces to a sequence of matrix-vector products, i.e.

- make \( y = U^Tb \),
- then \( z = S^{-1}y \),
- and \( x = Vz \).

Note that S is a diagonal matrix, so \( S^{-1} \) is promptly obtained.

The theoretical basis for the computation of the SVD is somewhat complicated and will not be discussed here (see [12]) for instance, which uses basically Householder transformations.

Once A is overwritten by U, it is required an additional space of order \( n \times n \) for the storage of matrix V. The diagonal matrix S obviously requires only a vector of dimension n for the storage of the non zero values (the singular values \( \sigma_i \)) of the main diagonal.

APPENDIX V

Condition number of a linear system

Let \( A \mathbf{x} = \mathbf{b} \) be a system of linear equations of order \( m \times n \) so that \( m \geq n \). It is important to know how much small perturbations in the elements of A or \( \mathbf{b} \) affect the solution vector x. This can be evaluated from the condition number of A, written as \( \kappa(A) \). If \( \kappa(A) \) is large, small perturbations in A or \( \mathbf{b} \) may cause significant perturbations in the solution vector x. In this case, A is said to be ill-conditioned.

The condition number is greater or equal to 1. It also indicates how much A is close to a non invertible matrix (see 9). Therefore, it is desirable that \( \kappa(A) \) be as low as possible.

If \( m = n \), it is possible to know (see [8,9,11]) that \( \kappa(A) = \sigma_{\text{max}}/\sigma_{\text{min}} \). In general, if \( \kappa(A) \neq 1 \), we use the SVD of A (see Appendix II) to obtain \( \kappa(A) = \sigma_{\text{max}}/\sigma_{\text{min}} \). The condition number defined in this way is known as the spectral condition number. Another important result is \( \kappa(A^T) = \kappa(A) \).

REFERENCES


BIOGRAPHIES

João Nelson Hoffmann was born in Lapa, state of Paraná, Brazil, on April 22, 1958. He received the B.S. degree in Electrical Engineering in 1981 and the M.S. degree in Applied and Computational Mathematics in 1993. Since 1988 he is with Companhia Paranaense de Energia (COPEL), Curitiba, Paraná, Brazil, working with high voltage transmission line researches.

Petrino Pulino was born in Bebedouro, state of São Paulo, Brazil, on October 14, 1956. He received the B.S. degree in Mathematics, the M.S. degree in Applied Mathematics and the Ph.D. degree in Electrical Engineering from University of Campinas (UNICAMP). Presently he is an assistant professor at the Department of Applied Mathematics of UNICAMP.